IN THE SPECIFICATION

Pursuant to 37 CFR § 1.121(b)(1)(i)-(ii), please delete the paragraph beginning on page 4, line 16 and continuing through line 29, and replace it with the following paragraph, which includes markings to show all the changes relative to the previous version of the paragraph:

"According to the present invention, an N+K coding technique is described for use to protect data that is being distributed in a redundant array of independent nodes (RAIN). The data itself may be of any type, and it may also include system metadata. According to the invention, the data to be distributed is encoded by a dispersal operation that uses a group of permutation ring operators. In a preferred embodiment, the dispersal operation is effected using a matrix of the form $[I_N\_C]$ where I_N is an n x n identity sub-matrix and C is a k x n sub-matrix of code blocks. The identity sub-matrix is used to preserve the original data. The sub-matrix C preferably comprises a set of permutation ring operators that are used to generate the code blocks. The operators are preferably "polynomials" that are selected from a group ring of a permutation group with base ring Z_2, e.g., a set of permutations whose action on the data is taken to be the XOR of the actions of the individual permutations The i^{th} code block is computed as: ~~$C_i = f(g_i1(A_1), \ldots g_in(A_n))$~~ $C_i = f(g_{i1}(A_1), \ldots g_{in}(A_n))$, where f() is preferably addition mod 2 (i.e., XOR), and g( ) is a permutation operator as described above. Each code block is preferably stored on a separate node."

Pursuant to 37 CFR § 1.121(b)(1)(i)-(ii), please delete the paragraph beginning on page 4, line 30 and continuing through page 5, line 10, and replace it with the following paragraph, which includes markings to show all the changes relative to the previous version of the paragraph:

"In a more specific embodiment, an N+K (4,2) coding scheme is implemented. In this case, a file to be archived comprises four (4) data blocks (A1, A2, A3, A4). A dispersal matrix comprises six (6) code blocks (C0, C1, C2, C3, C4, C5). Because the

identity sub-matrix is used, however, the first four code blocks (C0, C1, C2, C3) are just copies of the first four data blocks, and these data blocks ~~8are~~ are then stored in four distinct nodes of the array. The sub-matrix C is then generated as follows. Assume that *g* is a permutation operator that comprises a polynomial of cyclic permutations, such as: $b_0 * c^0 + b_1 * c^1 + b_k c^k + \ldots b_{(m-1)} * c^{(m-1)}$, where b_k is a bit (0 or 1), c^0 is the identity ("do nothing to the data"), and c^k is a cycle operation c repeated k times, e.g., the operation: "cycle the data k words." The i^{th} code block is then computed as: ~~$C_i = f(g_i1(A_1), \ldots g_in(A_n))$~~ $C_i = f(g_{i1}(A_1), \ldots g_{in}(A_n))$. The C4 code block is then stored in the 5^{th} node, and the C5 code block is stored in the 6^{th} node to complete the encoding process."

Pursuant to 37 CFR § 1.121(b)(1)(i)-(ii), please delete the paragraph beginning on page 9, line 22 and continuing through page 10, line 6, and replace it with the following paragraph, which includes markings to show all the changes relative to the previous version of the paragraph:

"Any scheme that reduces r from (c-1)(n-1) of the "copying" scheme involves using code blocks that somehow mix the data. In general, any such mixing can be thought of in terms of a matrix product such as illustrated in Figure 2. In this example, *G* is a *t* by *n* matrix, and *A* is an *n* column vector (the data blocks). This matrix product produces t (= n+k+r) code blocks. The i^{th} code block is computed as: ~~$C_i = f(g_i1(A_1), \ldots g_i n(A_n))$~~ $C_i = f(g_{i1}(A_1), \ldots g_{in}(A_n))$. Here, the *g*'s can be any functions acting on the code blocks, and *f* a function that acts on these intermediate results to mix them, producing the C_i elements. If this were a normal matrix computation over integers, the action of g_{ij} is a multiplication and the action of *f* is an addition. There is no reason, however, that these operations must be normal multiplications and additions. If one considers that the individual blocks are usually themselves long strings of information that may be mixed by the operators g_{ij}, each of the g_{ij} can itself be considered as a matrix, operating at a finer resolution. Similarly, one may move from any (non-block) erasure code to a block erasure code just by grouping the fine scale operations into blocks. Of course, designing the code to take

advantage of the block nature of the problem may produce computational savings. Decoding proceeds by inversion in the usual manner."

Pursuant to 37 CFR § 1.121(b)(1)(i)-(ii), please delete the paragraph beginning on page 10, line 4 and continuing through line 17, and replace it with the following paragraph, which includes markings to show all the changes relative to the previous version of the paragraph:

"Thus, according to the present invention, an NK coder is described for use to protect data that is being distributed in an RAIN archive. The data itself may be of any type, and it may also include the archive metadata. According to the invention, the data to be distributed is encoded by a matrix operation that uses an identity sub-matrix to preserve the data words, and that uses permutation ring operators to generate the code words. The operators are preferably polynomials that are selected from a group ring of a permutation group with base ring Z_2. The i^{th} code block is computed as: ~~$C_i = f(g_i1(A_1), \ldots g_in(A_n))$~~ $C_i = f(g_{i1}(A_1), \ldots g_{in}(A_n))$, where f( ) is preferably addition mod 2 (i.e., XOR), and g( ) is a permutation operator, such as a polynomial of cyclic permutations. Illustrative operators include, for example, $1 = s^0$ ("do nothing"), s^n ("shift right n words"), $1+s^n$ (XOR, unshifted image with shifted n), and so forth. With these operators, $(1+s)(a_1a_2a_3) = (a_1+a_3)(a_2+a_1)(a_3+a_2)$. The invention is desirable as most operators are very fast. Where matrices are not invertible, the de-convolve operation can be used, i.e., given a first word a1, decode $(1+s)(A) = (((a_1+a_3)(a_2+a_1)(a_3+a_2)$. A de-convolution example is shown in Figure 3."